Différences Temporelles de Kalman : le cas stochastique

Matthieu Geist[1,2,3], Olivier Pietquin[1] et Gabriel Fricout[2]

[1] Supélec
Equipe IMS, Metz, France
{matthieu.geist,olivier.pietquin}@supelec.fr

[2] ArcelorMittal Research
Cluster MC, Maizières-lès-Metz, France
gabriel.fricout@arcelormittal.com

[3] INRIA Nancy - Grand Est
Equipe-projet CORIDA, Nancy, France

Résumé : Les différences temporelles de Kalman (KTD pour *Kalman Temporal Differences*) sont un cadre de travail statistique qui traite de l'approximation de la fonction de valeur et de qualité en apprentissage par renforcement. Son principe est d'adopter une représentation paramétrique de la fonction de valeur, de modéliser les paramètres associés comme des variables aléatoires et de minimiser l'espérance de l'erreur quadratique moyenne des paramètres conditionnée à l'ensemble des récompenses observées. Ce paradigme s'est montré efficace en terme d'échantillons (*i.e.* convergence rapide), capable de prendre en compte la non-stationnarité ainsi que de fournir une information d'incertitude. Cependant ce cadre de travail était restreint au processus décisionnels de Markov bénéficiant de transitions déterministes. Dans cette contribution nous proposons d'étendre le modèle au transitions stochastiques à l'aide d'un bruit coloré, ce qui mène aux différences temporelles de Kalman étendues (XKTD pour *eXtended KTD*). L'approche proposée est illustrée sur des problèmes usuels en apprentissage par renforcement.

1 Introduction

Cet article aborde le problème de la détermination de la politique optimale d'un processus décisionnel de Markov (PDM) dans un contexte d'apprentissage par renforcement (AR), c'est-à-dire principalement apprentissage en ligne du contrôle optimal sans connaissance *a priori* du modèle. Un algorithme d'AR devrait posséder certaines caractéristiques : pouvoir prendre en compte de grands espaces d'état et d'action (approximation de la fonction de valeur), être efficace en terme d'échantillons (apprendre un bon contrôle avec aussi peu d'interactions que possible), prendre en compte la non-stationnarité (même si le système est stationnaire, le contrôler tout en cherchant à apprendre la politique optimale peut induire des non-stationnarités) et prendre en compte l'incertitude (qui est une condition quasi-nécessaire pour pouvoir s'intéresser au dilemme entre exploration et exploitation). Tous ces aspects font partie intégrante du cadre de travail des différences temporelles de Kalman (KTD pour *Kalman Temporal Differences*) de Geist *et al.* (2009a,b) que nous décrivons brièvement par la suite. Un algorithme d'AR devrait aussi pouvoir prendre en compte des transitions stochastiques, ce qui est l'objet de cette contribution.

Un PDM est un tuple $\{S, A, P, R, \gamma\}$ où S est l'espace d'état, A l'espace d'action, $p : s, a \in S \times A \rightarrow p(.|s, a) \in \mathcal{P}(S)$ une famille de probabilités de transitions, $R : S \times A \times S \rightarrow \mathbb{R}$ une fonction de récompense bornée et $\gamma \in [0, 1)$ un facteur d'actualisation. Une politique π associe à chaque état une probabilité sur les actions : $\pi : s \in S \rightarrow \pi(.|s) \in \mathcal{P}(A)$. La fonction de valeur d'une politique donnée associe à chaque état le cumul espéré de récompenses en partant de cet état et en suivant la politique par la suite, c'est-à-dire plus formellement $V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi]$ où r_i est la récompense immédiate observée au temps i, et l'espérance dépend des probabilités des trajectoires partant de s, étant données la dynamique du système et la politique suivie. La

Q-fonction (ou fonction de qualité) ajoute un degré de liberté supplémentaire sur le choix de la première action, et elle est définie par $Q^\pi(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi]$. L'objectif de l'AR est de déterminer (à partir d'interactions) la politique π^* qui maximise la fonction de valeur pour chaque état : $\pi^* = \text{argmax}_\pi(V^\pi)$. A la base, le cadre de travail de KTD ne permet de traiter que les PDM aux transitions déterministes, ce qui sera le cas de la première partie de cet article : les transitions et politiques deviennent déterministes, et nous les notons pour l'instant $p(s,a)$ et $\pi(s)$.

Deux schémas (plus généraux que le cas déterministe) parmi d'autres peuvent mener à la politique optimale. Premièrement, l'*itération de la politique* implique d'apprendre la fonction de valeur d'une politique donnée, puis d'améliorer cette politique, la nouvelle étant gloutonne respectivement à la fonction de valeur apprise. Cela implique de résoudre l'*équation d'évaluation de Bellman*, qui est donnée ici pour l'évaluation de la fonction de valeur ainsi que de la Q-fonction :

$$V^\pi(s) = R(s, \pi(s), s') + \gamma V^\pi(s'), \forall s \tag{1}$$

$$Q^\pi(s,a) = R(s,a,s') + \gamma Q^\pi(s', \pi(s')), \forall s, a \tag{2}$$

où s' dénote l'état vers lequel transite le système, c'est-à-dire $s' = p(s, \pi(s))$ ou $s' = p(s,a)$, selon le contexte. Le second schéma, l'*itération de la valeur*, permet de directement déterminer la politique optimale. Il implique de résoudre l'*équation d'optimalité de Bellman* :

$$Q^*(s,a) = R(s,a,s') + \gamma \max_{b \in A} Q^*(s', b), \forall s, a \tag{3}$$

Un important problème en AR est de trouver une solution approchée de l'une des équations de Bellman lorsque l'espace d'état et/ou d'action est trop grand pour les algorithmes classiques d'AR ou de programmation dynamique (Bertsekas & Tsitsiklis, 1996).

Pour ce faire, les méthodes de différences temporelles (TD pour *Temporal Differences*) sont considérées. Elle forment une classe d'approches qui consistent à corriger une représentation de la fonction de valeur (ou de qualité) en accord avec l'erreur de différence temporelle définie ci-après. La plupart d'entre elles peuvent s'écrire de façon générique comme :

$$\theta_i = \theta_{i-1} + K_i \delta_i \tag{4}$$

Dans cette expression, θ_{i-1} est la dernière représentation de la fonction de valeur (un vecteur de paramètres), θ_i est une mise à jour de cette représentation étant donnée une transition observée, δ_i est l'erreur dite de différence temporelle (la différence entre les membre de droite et de gauche d'une des équations de Bellman pour une transition échantillonnée), et K_i est un gain qui indique dans quelle direction la représentation de la fonction de valeur doit être corrigée.

Par exemple, des algorithmes comme TD, SARSA ou encore Q-learning avec approximation de la valeur (Sutton & Barto, 1998) peuvent être écrits de façon générique sous la forme (4). Pour tous ces algorithmes le vecteur de paramètres dépend de la paramétrisation choisie (réseaux de neurones, machines à noyaux, *tile coding*, tabulaire, *etc*). Pour TD, qui est lié à l'équation (1), l'erreur TD est $\delta_i = r_i + \gamma \hat{V}_{\theta_{i-1}}(s_{i+1}) - \hat{V}_{\theta_{i-1}}(s_i)$, i étant l'indice temporel. Pour SARSA, qui est lié à (2), cette erreur est $\delta_i = r_i + \gamma \hat{Q}_{\theta_{i-1}}(s_{i+1}, a_{i+1}) - \hat{Q}_{\theta_{i-1}}(s_i, a_i)$. Pour le Q-learning, lié à (3), c'est $\delta_i = r_i + \gamma \max_{b \in A} \hat{Q}_{\theta_{i-1}}(s_{i+1}, b) - \hat{Q}_{\theta_{i-1}}(s_i, a_i)$. Pour TD le gain est $K_i = \alpha_i \nabla_{\theta_{i-1}} \hat{V}_{\theta_{i-1}}(s_i)$, où α_i est un taux d'apprentissage, et pour SARSA et Q-learning ce gain vaut $K_i = \alpha_i \nabla_{\theta_{i-1}} \hat{Q}_{\theta_{i-1}}(s_i, a_i)$. Geist *et al.* (2009a,b) montrent comment d'autres algorithmes de l'état de l'art comme TD(λ) (Sutton & Barto, 1998), LSTD (Bradtke & Barto, 1996) ou encore les algorithmes dits résiduels (Baird, 1995) peuvent être génériquement exprimés sous la forme de l'équation (4).

En postulant une mise à jour linéaire de la forme (4), le principe de KTD est de modéliser le vecteur de paramètres comme un vecteur aléatoire (éventuellement non-stationnaire) et de minimiser l'espérance de l'erreur quadratique sur les paramètres conditionnée aux récompenses passées, ce qui définit un gain K_i spécifique, le gain de Kalman. Cela donne naissance à des algorithmes du second ordre. Ce cadre de travail permet également de prendre en compte les non-stationnarités (le filtrage de Kalman ayant été développé à l'origine pour traquer l'état caché d'un système dynamique et stochastique). Les paramètres sont modélisés comme des variables aléatoires, ce qui permet d'estimer une information d'incertitude sur les valeurs approchées, comme illustré par Geist *et al.* (2009a). De plus, cette approche est en ligne, ce qui est un aspect de l'AR que nous pensons important. La

section 2 présente le cadre de KTD dans le cas des transitions déterministes. Il est ensuite étendu au cas des transitions stochastiques en utilisant un modèle de bruit proposé à l'origine par Engel *et al.* (2005) dans un contexte différent. La section 4 montre une série d'expérimentations, et la dernière section ouvre quelques perspectives.

2 Différences temporelles de Kalman

Le paradigme de KTD est brièvement décrit dans cette section, pour plus de détails théoriques et une comparaison expérimentale à l'état de l'art le lecteur peut se référer à Geist *et al.* (2009a). Une transition est noté de façon générique par :

$$t_i = \begin{cases} (s_i, s_{i+1}) & \text{(5a)} \\ (s_i, a_i, s_{i+1}, a_{i+1}) & \text{(5b)} \\ (s_i, a_i, s_{i+1}) & \text{(5c)} \end{cases}$$

étant donné que l'objectif est l'évaluation de la fonction de valeur (5a), l'évaluation de la Q-fonction (5b) ou encore l'optimisation directe de la fonction de qualité (5c). De façon similaire, pour les mêmes cas, les notations suivantes sont adoptées :

$$g_{t_i}(\theta_i) = \begin{cases} \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) & \text{(6a)} \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) & \text{(6b)} \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_b \hat{Q}_{\theta_i}(s_{i+1}, b) & \text{(6c)} \end{cases}$$

Le lecteur remarquera que l'équation $r_i = g_{t_i}(\theta)$ correspond aux équations de Bellman pour une transition donnée. Un point de vue statistique est adopté. Le problème est exprimé dans une formulation dite *espace-d'état* :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i \\ r_i = g_{t_i}(\theta_i) + n_i \end{cases} \tag{7}$$

En utilisant le vocabulaire du filtrage de Kalman (1960), la première équation est l'équation d'évolution. Elle spécifie que le vecteur de paramètres suis une marche aléatoire dont l'espérance correspond à l'estimation optimale de la fonction de valeur. Par définition, le bruit d'évolution v_i est centré, blanc, indépendant et de matrice de variance P_{v_i}. La seconde équation est l'équation d'observation, elle lie la transition observée (ainsi que la récompense immédiate) à la fonction de valeur (ou de qualité) à l'aide d'une des équations de Bellman (1-3). Le bruit d'observation n_i est supposé centré, blanc, indépendant et de variance P_{n_i}. Ce modèle de bruit vient du fait que la solution de Bellman n'existe pas nécessairement dans l'espace fonctionnel engendré par l'ensemble des paramètres (la structure de l'approximateur étant fixée *a priori*, le choix de l'approximateur étant un problème en soit que nous n'abordons pas dans cette contribution).

Cette formulation espace-d'état postule que les récompenses sont générées en accord avec l'équation d'observation, qui est elle même conduite par les paramètres aléatoires cachés qui définissent une famille de fonctions dont l'espérance correspond à la fonction de valeur approchée optimale. C'est un modèle statistique de l'équation (approchée) de Bellman, et l'équation d'évolution *n'est pas* une équation de mise à jour. De plus, si le problème d'intérêt est formellement stationnaire, il n'y a pas de bruit d'évolution. Cependant, le modèle de marche aléatoire permet de prendre en compte les non-stationnarités, par exemple induites par la dualité entre apprentissage et contrôle, comme montré par Geist *et al.* (2009a), et cela sans compromettre le cas stationnaire. De plus, empiriquement, un bruit artificiel d'évolution peut permettre d'éviter les minima locaux. De façon générale nous introduisons donc un bruit d'évolution, même s'il est artificiel.

L'objectif de KTD est la minimisation de l'espérance de l'erreur quadratique conditionnée aux observations (transitions) passées :

$$J(\hat{\theta}_i) = E\left[\|\theta_i - \hat{\theta}_i\|^2 | r_{1:i}\right] \text{ avec } r_{1:i} = r_1, \ldots, r_i \tag{8}$$

De façon générale, l'estimateur minimisant ce coût est l'espérance conditionnelle. Cependant, à part dans certains cas bien spécifiques (par exemple linéaire et gaussien), cet estimateur ne peut

pas être calculé. A la place, l'objectif est de trouver le meilleur estimateur *linéaire*, qui peut être écrit sous une forme très similaire à l'équation (4). Minimiser ce coût sous cette hypothèse donne naissance à l'approche la plus générale de KTD, qui est résumée dans l'algorithme 1, pour lequel les notations suivantes sont utilisées : $\hat{\theta}_{i|j} = E[\theta_i|r_{1:j}]$ est la prédiction de l'estimateur au temps i conditionnée aux récompenses $r_{1:j}$ observées jusqu'au temps j, et $P_{i|j} = \text{cov}(\theta_i - \hat{\theta}_{i|j}|r_{1:j})$ est la matrice de variance associée. Une chose importante à noter est que la dérivation de cet algorithme ne fait pas d'hypothèse gaussienne et n'utilise pas directement la règle de Bayes (même si le cadre de travail proposé a des connexions avec l'apprentissage bayésien[1]).

L'algorithme KTD se décompose en trois étapes. Premièrement, l'étape de prédiction consiste à mettre à jour l'espérance et la variance du vecteur de paramètres en accord avec l'équation d'évolution. Ensuite certaines statistiques d'intérêt sont calculées. Elles sont nécessaire pour l'étape de correction (la troisième étape), qui consiste à corriger les moments d'ordre un et deux du vecteur de paramètres en fonction du gain de Kalman K_i, de la récompense prédite $\hat{r}_{i|i-1}$ et de la récompense observée r_i. La différence entre récompenses prédite et observée, appelée innovation dans le paradigme du filtrage de Kalman, est une forme d'erreur de différence temporelle. Il est à noter qu'étant en ligne, KTD doit être initialiser avec un *a priori* sur les paramètres moyens et la matrice de variance associée.

Le problème est que dans le cas général les statistiques d'intérêt ne sont pas calculables, excepté pour le cas linéaire, qui ne tient pas si la paramétrisation est non linéaire (réseau de neurones par exemple) ou pour l'équation d'optimalité de Bellman, à cause de l'opérateur max. Cependant, un schéma d'approximation ne requérant pas de calcul de gradient, la transformation non-parfumée de Julier & Uhlmann (2004) (UT pour *Unscented Transform*), permet d'estimer les moments d'ordre un et deux de la transformation non-linéaire d'une variable aléatoire en connaissant moyenne et variance de la variable aléatoire non-transformée. Cela est utilisé pour dériver une famille d'algorithmes, nommément KTD-V (6a), KTD-SARSA (6b) et KTD-Q (6c), selon que l'objectif est l'évaluation de la fonction de valeur, de qualité, ou l'optimisation directe de la Q-fonction. Les détails et développements peuvent être trouvés chez Geist *et al.* (2009b).

Algorithme 1 : Algorithme KTD général.

Initialisation : *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i et la récompense r_i ;

Etape de prédiction;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des statistiques d'intérêt (en utilisant l'UT et les statistiques $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$);
$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i)|r_{1:i-1}]$;
$P_{\theta r_i} = E\left[(\theta_i - \hat{\theta}_{i|i-1})(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})|r_{1:i-1}\right]$;
$P_{r_i} = E\left[(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})^2|r_{1:i-1}\right] + P_{n_i}$;

Phase de correction;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

[1] De façon générale, le filtrage de Kalman peut être vu sous trois optiques différentes : projection orthogonale, filtrage bayésien (approché si non-linéaire ou non-gaussien) et statistique. C'est ce dernier point de vue que nous adoptons.

3 Transitions stochastiques

Supposons à présent que les transitions soient stochastiques (les politiques sont toujours supposées déterministes). Nous nous concentrons dans un premier temps sur l'évaluation de la fonction de valeur. L'extension à l'évaluation de la Q-fonction est simple, et l'optimisation directe de la fonction de qualité pose des problèmes particuliers à cause de son aspect *off-policy* (la politique apprise n'est pas la politique suivie) que nous discutons par la suite. L'équation de Bellman qu'il s'agit maintenant de résoudre est l'espérance de l'équations (1) :

$$V^\pi(s) = E_{s'|s,\pi(s)}\left[R(s,\pi(s),s') + \gamma V^\pi(s')\right], \forall s \tag{9}$$

Il peut être montré (voir Geist *et al.* (2009a)) qu'utiliser directement KTD dans un problème stochastique induit un biais de la fonction de coût minimisée (8), ce biais étant très similaire à celui apparaissant lors de la minimisation d'un résidu quadratique de Bellman, comme explicité par Antos *et al.* (2008). Nous rappelons l'expression de ce biais :

$$\text{trace}\left(K_i E\left[\text{cov}_{s'|s_i}\left(r_i + \gamma V_\theta(s')\right)|r_{1:i-1}\right] K_i^T\right) = \|K_i\|^2 E\left[\text{cov}_{s'|s_i}\left(r_i + \gamma V_\theta(s')\right)|r_{1:i-1}\right] \tag{10}$$

où K_i est le gain de Kalman, $\|.\|$ est la norme euclidienne usuelle, la covariance dépend des probabilités de transition et l'espérance est sur les paramètres conditionnés aux observations passées. De plus, sous certaines hypothèses (bruits et *a priori* gaussien, canal sans mémoire), l'estimateur de KTD (en utilisant la transformation non-parfumée) est l'estimateur du maximum *a posteriori* (voir van der Merwe (2004, chapitre 4.5) pour une preuve dans le cas général d'un filtre de Kalman à sigma-points dont le modèle d'évolution est une marche aléatoire). Il est possible de montrer que cet estimateur du maximum *a posteriori*, sous ces mêmes hypothèses, minimise le coût empirique suivant :

$$C_i(\theta) = \sum_{j=1}^{i} \frac{1}{P_{n_j}} \left(r_j - g_{t_j}(\theta)\right)^2 \tag{11}$$

Sous cette forme, le lien avec les problèmes liés aux transitions stochastiques des approches minimisant un résidu quadratique de Bellman est encore plus clair. Cette contribution propose d'étendre KTD avec un modèle de bruit coloré ayant été introduit par Engel *et al.* (2005) pour une approche bayésienne basée sur une modélisation de la fonction de valeur par un processus gaussien.

3.1 Un modèle de bruit coloré

La politique étant fixée dans un contexte d'évaluation, le PDM se réduit à une chaîne de Markov valuée dont la probabilité de transition est donnée par $p^\pi(.|s) = p(.|s,\pi(s))$ et dont la récompense est $R^\pi(s,s') = R(s,\pi(s),s')$. La fonction de valeur peut être définie comme l'espérance (sur l'ensemble des trajectoires possibles) du processus aléatoire du cumul pondéré de récompenses suivant :

$$D^\pi(s) = R^\pi(s,s') + \gamma D^\pi(s'),\ s' \sim p^\pi(.|s) \tag{12}$$

Ce processus aléatoire peut se décomposer en deux parties, la fonction de valeur et un résiduel aléatoire centré :

$$D^\pi(s) = V^\pi(s) + \Delta V^\pi(s) \tag{13}$$

où par définition $V^\pi(s) = E[D^\pi(s)]$ et $\Delta V^\pi(s) = D^\pi(s) - V^\pi(s)$ est le résidu. En injectant l'équation (13) dans l'équation (12), la récompense peut être exprimée comme une fonction de la valeur plus un bruit :

$$R^\pi(s,s') = V^\pi(s) - \gamma V^\pi(s') + N(s,s') \tag{14}$$

le bruit étant défini par :

$$N(s,s') = \Delta V^\pi(s) - \gamma \Delta V^\pi(s') \tag{15}$$

Comme Engel *et al.* (2005), nous supposons les résidus indépendants, ce qui mène à un modèle de bruit coloré.

3.2 Extension de KTD

Rappelons l'équation d'observation de la formulation espace-d'état (7) : $r_i = g_{t_i}(\theta_i) + n_i$. Dans le cadre de travail de KTD, le bruit d'observation n_i est supposé blanc, ce qui est nécessaire à l'obtention de l'algoritme final. Dans la version étendue de KTD (XKTD pour *eXtended Kalman Temporal Differences*), le modèle de bruit coloré (15) est utilisé à la place.

Les résidus étant centrés et supposés indépendants, ce bruit est en fait un bruit à moyenne mobile[2] (bruit MA pour *Moving Average*) qui est la somme de deux bruits blancs :

$$n_i = -\gamma u_i + u_{i-1}, \quad u_i \sim (0, \sigma_i^2) \tag{16}$$

Notons que le bruit blanc u_i est centré de variance σ_i^2, cependant aucune supposition n'est faite à propose de sa distribution (et particulièrement pas d'hypothèse gaussienne). S'il est assez aisé d'utiliser un modèle de bruit d'observation auto-régressif[3] (AR pour *auto-regressive*) en étendant l'équation d'évolution (voir Simon (2006) par exemple), le cas d'un bruit d'observation MA n'a jamais été traité dans la littérature, autant que nous le sachions.

Redériver KTD dans le cas d'un bruit MA serait bien trop difficile. A la place, nous proposons d'exprimer le bruit MA scalaire n_i comme un bruit AR vectoriel. Cela permet d'étendre le modèle espace-d'état (7) à un nouveau pour lequel l'algorithme 1 s'applique assez directement. Soit w_i une variable aléatoire auxiliaire. Le bruit MA scalaire (16) est équivalent au bruit AR vectoriel suivant :

$$\begin{pmatrix} w_i \\ n_i \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} w_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ -\gamma \end{pmatrix} u_i \tag{17}$$

Le bruit $u'_i = \begin{pmatrix} u_i & -\gamma u_i \end{pmatrix}^T$ est également centré et sa matrice variance est donnée par :

$$P_{u'_i} = \sigma_i^2 \begin{pmatrix} 1 & -\gamma \\ -\gamma & \gamma^2 \end{pmatrix} \tag{18}$$

Cette nouvelle formulation du bruit d'observation ayant été définie, il est maintenant possible d'étendre la formulation espace-d'état (7) :

$$\begin{cases} \mathbf{x}_i = F\mathbf{x}_{i-1} + v'_i \\ r_i = g_{t_i}(\mathbf{x}_i) \end{cases} \tag{19}$$

Le vecteur de paramètres est maintenant étendu avec le bruit AR vectoriel $\begin{pmatrix} w_i & n_i \end{pmatrix}^T$:

$$\mathbf{x}_i^T = \begin{pmatrix} \theta_i^T & w_i & n_i \end{pmatrix} \tag{20}$$

La matrice d'évolution F prend en compte la structure du bruit d'observation MA (sous sa forme AR). Soit p le nombre de paramètres et I_p la matrice identité de taille p, la matrice d'évolution s'écrit par bloc ($\mathbf{0}$ étant un vecteur colonne de taille $p \times 1$) :

$$F = \begin{pmatrix} I_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & 0 & 0 \\ \mathbf{0}^T & 1 & 0 \end{pmatrix} \tag{21}$$

Le bruit d'évolution v_i est également étendu pour prendre en compte le bruit d'observation coloré. Il est toujours centré, cependant la matrice de variance est maintenant définie par :

$$P_{v'_i} = \begin{pmatrix} P_{v_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \sigma_i^2 & -\gamma\sigma_i^2 \\ \mathbf{0}^T & -\gamma\sigma_i^2 & \gamma^2\sigma_i^2 \end{pmatrix} \tag{22}$$

[2]De façon générale, un bruit à moyenne amovible y_k est défini par $y_k = \sum_{j=0}^{q} b_j u_{k-j}$ où $(u_k)_k$ est un bruit blanc et $b_0, \ldots, b_q$ est un ensemble de coefficients.

[3]De façon générale, un bruit auto-régressif y_k est défini par $y_k = \sum_{j=1}^{q} a_i y_{k-i} + u_k$ où $(u_k)_k$ est un bruit blanc et $a_1, \ldots, a_q$ sont des coefficients.

L'équation d'observation reste la même :

$$r_i = g_{t_i}(\mathbf{x}_i) = g_{t_i}(\theta_i) + n_i \tag{23}$$

Cependant maintenant le bruit d'observation fait partie intégrante de l'équation d'évolution, au même titre que les paramètres.

En utilisant cette nouvelle formulation espace-d'état, l'algorithme général XKTD peut être dérivé. Il est résumé dans l'algorithme 2, qui est très similaire à l'algorithme 1. La seule chose qui change (excepté le fait que le modèle espace d'état ne soit pas le même) est l'étape de prédiction : la prédiction de la moyenne et de la variance du vecteur aléatoire étendu $\mathbf{x}_i$ est faite en utilisant la matrice d'évolution F (cette matrice étant l'identité pour KTD claissique). Notons que le coût computationnel est le même pour KTD (coût quadratique, voir Geist *et al.* (2009a) pour les détails) et son extension XKTD, étant donné que le vecteur de paramètres est étendu avec seulement deux variables. Comme pour KTD, XKTD peut être spécialisé en XKTD-V (évaluation de la fonction de valeur) et XKTD-SARSA (évaluation de la fonction de qualité). Le raisonnement est exactement le même que celui développé par Geist *et al.* (2009b) et n'est pas répété ici. La spécialisation à XKTD-Q n'est pas évidente en raison de son aspect *off-policy*.

Algorithme 2 : Algorithme XKTD général.

Initialisation : *a priori* $\hat{\mathbf{x}}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i et la récompense r_i ;

Phase de prédiction;
$\hat{\mathbf{x}}_{i|i-1} = F\hat{\mathbf{x}}_{i-1|i-1}$;
$P_{i|i-1} = FP_{i-1|i-1}F^T + P_{v'_i}$;

Calcul des statistiques d'intérêt (en utilisant l'UT et les statistiques $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$);
$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i) + n_i | r_{1:i-1}]$;
$P_{\mathbf{x}r_i} = E\left[(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1}) | r_{1:i-1}\right]$;
$P_{r_i} = E\left[(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1})^2 | r_{1:i-1}\right]$;

Phase de correction;
$K_i = P_{\mathbf{x}r_i} P_{r_i}^{-1}$;
$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

3.3 XKTD pondéré

Dans cette section, une version pondérée de XKTD est proposée. Elle a été pensée à l'origine pour traiter le problème causé par un apprentissage *off-policy*, et nous verrons en quoi elle ne peut prendre en compte que partiellement ce problème. Cependant, l'approche proposée est également valable pour des politiques stochastiques dans le cadre d'un apprentissage *on-policy*, ce qui est montré être particulièrement utile pour un apprentissage ϵ-glouton dans la section 4.

L'apprentissage *off-policy* est le problème d'apprendre une politique (la politique cible) tout en en suivant une autre (la politique comportementale). KTD-Q (et plus généralement les algorithmes qui s'inscrivent dans le cadre d'un schéma d'itération de la valeur, comme Q-learning) est un exemple d'apprentissage *off-policy* : la politique comportementale est n'importe quelle politique suffisamment explorative tandis que la politique cible est la politique optimale. Plus généralement, l'apprentissage *off-policy* est d'intérêt, par exemple pour réutiliser des trajectoires déjà observées ou encore si la politique comportementale ne peut pas être contrôlée.

L'utilisation d'un bruit d'observation coloré résulte en un effet mémoire, de façon similaire à ce qu'il se passe pour les traces d'éligibilité pour les algorithmes plus classiques de différences temporelles (Sutton & Barto, 1998). Pour une paramétrisation linéaire et un bruit d'évolution nul (c'est-à-dire $P_{v_i} = 0$), l'algorithme XKTD-V est le même que la version paramétrique de

l'algorithme MC-GPTD de Engel (2005, chapitre 4.4), qui est obtenu en utilisant le même bruit coloré, des processus gaussiens et des arguments bayésiens. Ce dernier algorithme est équivalent à estimer la fonction de valeur de façon supervisée, les cibles étant des échantillons obtenus par Monte-Carlo du cumul pondéré de récompenses. La cible de l'algorithme LSTD(1) est la même, comme noté par Engel (2005, chapitre 4.4).

Comme les algorithme plus classiques basés sur les traces d'éligibilité, XKTD appliqué à de l'apprentissage *off-policy* est susceptible d'échouer car il inclue des effets de toute la trajectoire, pas seulement de la transition courante, ces effets étant contaminés par la politique comportementale et n'étant compensés d'aucune façon. Cependant on peut envisager de les prendre en compte dans le modèle de bruit. Considérons le bruit (15) pour la fonction de qualité :

$$N(s_i, a_i, s_{i+1}, a_{i+1}) = \Delta Q(s_i, a_i) - \gamma \Delta Q(s_{i+1}, a_{i+1}) \tag{24}$$

Soit $(s_i, a_i, s_{i+1}, a_{i+1}, s_{i+2}, a_{i+2})$ une partie de la trajectoire générée en accord avec la politique comportementale b. Supposons que la politique cible est π, la mise à jour est donc faite selon $(s_i, a_i, s_{i+1}, a^{\pi}_{i+1})$ où l'action a^{π}_{i+1} est choisie en accord avec la politique π. La variance du bruit est toujours la même, cependant la corrélation entre les bruits à deux instants consécutifs est maintenant :

$$E[N(s_i, a_i, s_{i+1}, a^{\pi}_{i+1}) N(s_{i+1}, a_{i+1}, s_{i+2}, a^{\pi}_{i+2})] = -\gamma \pi(a_{i+1}|s_{i+1}) E[(\Delta Q(s_{i+1}, a_{i+1}))^2] \tag{25}$$

La corrélation est donc pondérée par la probabilité de l'action a_{i+1} d'avoir été échantillonnée par la politique π. Cela mène à une légère modification de l'algorithme XKTD. Le changement se fait dans la variance (18) du bruit auto-régressif vectoriel :

$$P_{u'_i} = \sigma_i^2 \begin{pmatrix} 1 & -\gamma\pi(a_{i+1}|s_{i+1}) \\ -\gamma\pi(a_{i+1}|s_{i+1}) & \gamma^2 \end{pmatrix} \tag{26}$$

Notons que pour l'apprentissage *on-policy* d'une politique déterministe nous retrouvons la mise à jour de XKTD. Nous appelons cette approche XKTD pondéré (ou wXKTD pour *weighted XKTD*).

Cet algorithme partage des similarités avec le Q(λ) de Watkins (1989) qui coupe les traces d'éligibilité dès qu'une action explorative est prise, ainsi qu'avec l'algorithme *tree backup* de Precup *et al.* (2000) qui pondère la trace par la probabilité de l'action choisie par la politique comportementale d'avoir été choisie par la politique cible, de façon similaire à l'approche proposée. Si la trace d'éligibilité est coupée pour l'un de ces algorithmes, la mise à jour se fait selon Q(0) et SARSA(0) qui sont des estimateurs non biaisés de la fonction de qualité. Cependant, si le bruit est coupé pour wXKTD, la mise à jour correspondante sera celle de KTD, qui est un estimateur biaisé de la Q-fonction si les transitions ne sont pas déterministes. Ainsi, dans le cas *off-policy*, l'approche proposée pourra peut être réduire le biais, mais pas complètement le supprimer. En fait, elle réalise un compromis entre biais causé par la stochasticité des transitions et biais causé par l'aspect *off-policy*.

Cependant, dans le cas d'un apprentissage *on-policy*, si la politique suivie est stochastique (par exemple ϵ-gloutonne dans le cadre d'un schéma d'itération optimiste de la politique), l'approche pondérée proposée est toujours valable et peut apporter des améliorations par rapport à XKTD. Cela est expérimenté dans la section 4.

4 Résultats expérimentaux

Dans cette section, un certain nombre d'expérimentations est proposé. La première illustre la réduction de biais lors de l'évaluation de la fonction de valeur d'une simple (mais non-stationnaire) chaîne de Markov valuée. Ensuite, le problème de l'évaluation de la Q-fonction est testé sur une chaîne contrôlée, ainsi que des algorithmes de type ϵ-glouton (aspect *on-policy* et politique stochastique) et itération de la valeur (aspect *off-policy*). Enfin les différentes variantes de KTD (ainsi que TD et LSTD) sont comparées sur une version stochastique du problème bien connu du *mountain-car*.

4.1 Chaîne de Boyan

La chaîne de Boyan (1999) est une chaîne de Markov à 13 états où l'état s_0 est absorbant de récompense nulle, s^1 transite vers s^0 avec une probabilité de 1 et une récompense de -2, et s^i transite vers s^{i-1} ou s^{i-2}, $2 \leq i \leq 12$, chaque transition ayant une probabilité de 0.5 et une récompense de -3. Pour cette expérimentation, (X)KTD-V est comparé à TD (Sutton & Barto, 1998) et LSTD (Bradtke & Barto, 1996). La paramétrisation est linéaire, et les vecteurs de base $\phi(s)$ pour les états s^{12}, s^8, s^4 et s^0 sont respectivement $[1,0,0,0]^T$, $[0,1,0,0]^T$, $[0,0,1,0]^T$ et $[0,0,0,1]^T$. Les vecteurs de base pour les autres états sont obtenus par interpolation linéaire. La fonction de valeur approchée est donc $\hat{V}_\theta(s) = \theta^T \phi(s)$. La fonction de valeur optimale est exactement linéaire en ces bases, et le vecteur optimal de paramètres correspondant est $\theta^*_{(\text{-})} = [-24, -16, -8, 0]^T$. Pour mesurer la qualité de chaque algorithme la distance euclidienne entre le vecteur de paramètres courant est l'optimal $\|\theta - \theta^*\|$ est utilisée.

La facteur d'actualisation γ est fixé à 1 pour cette tâche épisodique. Pour TD, le taux d'apprentissage est fixé à $\alpha = 0.1$. Pour LSTD l'*a priori* est fixé à $P_{0|0} = I$ où I est la matrice identité. Pour (X)KTD-V le même *a priori* est utilisé, la variance des résidus (resp. la variance du bruit d'observation) est fixée à $\sigma_i^2 = 10^{-3}$ (resp. $P_{n_i} = 10^{-3}$) et le bruit d'évolution est adaptatif, la variance associée est de la forme $P_{v_i} = \eta P_{\theta_{i-1|i-1}}$ où $P_{\theta_{i-1|i-1}}$ est égal à $P_{i-1|i-1}$ pour KTD et au bloc haut gauche de taille $p \times p$ de cette matrice pour XKTD, et $\eta \ll 1$ est une constante positive, choisie ici égale à 10^{-2}. Ce type de bruit artificiel, assez classique dans la littérature du traitement de signal, met l'emphase sur les données les plus récentes. Choisir ces paramètres requiert un peu de pratique, mais pas plus que de choisir un taux d'apprentissage pour d'autres algorithmes. Pour tous les algorithmes le vecteur de paramètres initial est choisi égal à zéro.

KTD a été pensé pour pouvoir prendre en compte les environnements non-stationnaires, et XKTD devrait également présenter cette caractéristique. Pour simuler un changement dans le PDM, le signe de la récompense est inversé à partir du 100$^{\text{ème}}$ épisode. La fonction de valeur optimale est toujours linéaire en les bases, et le vecteur optimal de paramètres correspondant est $\theta^*_{(+)} = -\theta^*_{(\text{-})}$ après le changement du PDM. L'apprentissage est fait sur 200 épisodes, et les résultats sont moyennées sur 100 essais (chaque essai correspond à 200 épisodes). Les résultats sont présentés sur la figure 1.

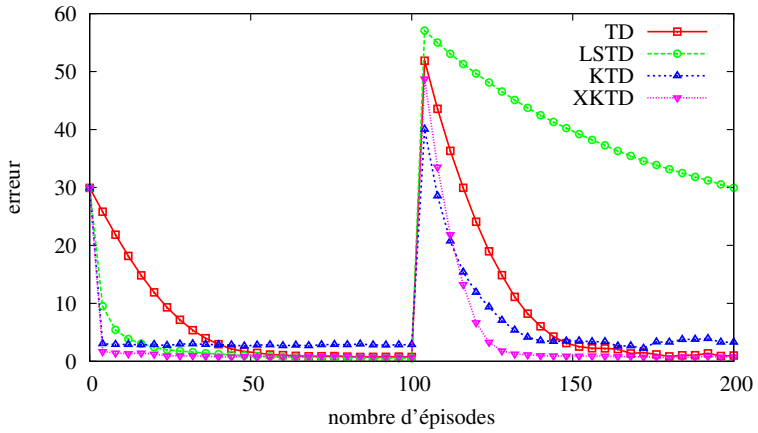


FIG. 1 – Chaîne de Boyan.

Avant le changement de PDM, KTD converge plus rapidement que LSTD, qui converge plus rapidement que TD. Cependant, comme prévu, KTD est biaisé. L'algorithme XKTD converge aussi rapidement que KTD, cependant sans être biaisé. Après le changement dans le PDM, les résultats sont similaires à ceux présentés par Geist *et al.* (2009a). LSTD échoue à s'adapter au nouveau PDM, TD fonctionne mieux grâce à sont taux d'apprentissage constant (un taux d'apprentissage ne décroissant pas trop vite aurait suffit) et KTD s'adapte encore plus rapidement. XKTD fait la même chose, le biais en moins. Notons que le choix du bruit a son importance. S'il y a trop de bruit d'évolution, l'adaptation sera rapide mais l'apprentissage sera relativement instable, particulièrement avec des transitions stochastiques (l'aspect aléatoire des transitions peut

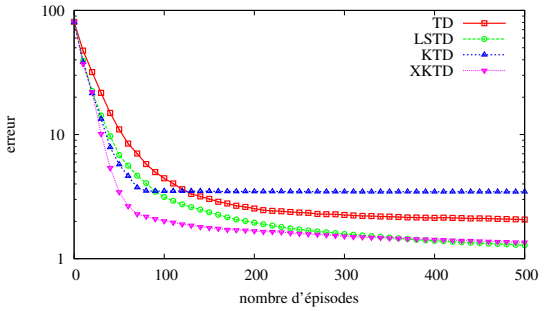
être alors interprétée comme un changement immédiat dans le PDM), et si le bruit est trop faible, l'adaptation sera lente. C'est une forme de dilemme entre plasticité et stabilité, et le même type de problème se pose en choisissant un taux d'apprentissage. Choisir un bruit d'évolution artificiel comme celui proposé réduit en partie ce problème.

4.2 Chaîne contrôlée

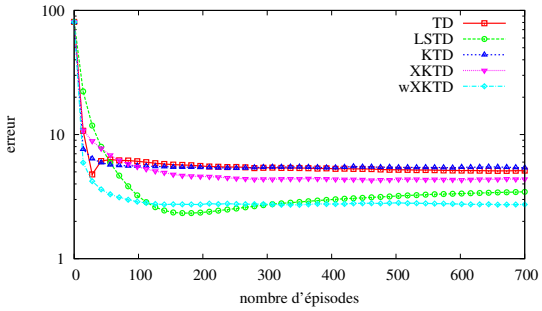
Le domaine utilisé dans cette section est un PDM à six états et deux actions. Les actions sont aller à gauche et à droite. Avec une probabilité $p = 0.9$ l'action choisie est effectuée, sinon son contraire est fait. La récompense est -3 dans chaque état sauf le dernier qui a une récompense de -2. La politique optimale est de systématiquement aller à droite. Une représentation tabulaire de la Q-fonction est choisie (c'est une paramétrisation possible parmi d'autre, justifiée pour ce simple PDM). Le facteur d'actualisation est fixé à $\gamma = 0.9$. Dans toutes les expériences la mesure de performance est la distance euclidienne entre la paramétrisation courante et la paramétrisation optimale $\|Q^* - \hat{Q}_\theta\|^2$.

4.2.1 Evaluation de la Q-fonction

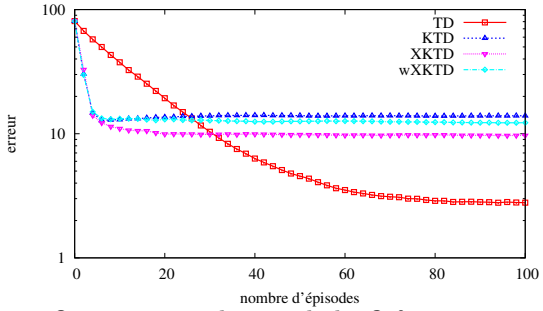
Nous nous intéressons dans un premier temps au problème de l'évaluation de la fonction de qualité. L'*a priori* pour LSTD et (X)KTD est $5I$. Pour TD, le taux d'apprentissage est fixé à $\alpha_i = \frac{n_0+1}{n_0+i}$ avec $n_0 = 500$. Il n'y a pas de bruit d'évolution, et la variance du bruit d'observation (resp. la variance des résidus) est fixée à 5.10^{-3} pour (X)KTD. L'apprentissage se fait sur 500 épisodes (chacun étant initialisé avec une paire état-action aléatoire et limité à 15 interactions), et les résultats présentés sur la figure 2.a sont moyennés sur 100 essais (notons l'échelle logarithmique). La politique suivie est la politique optimale. Il apparaît que KTD et XKTD convergent plus rapidement que TD et LSTD. KTD est biaisé, contrairement à XKTD. Cela confirme les résultats de la section 4.1.



a. Evaluation de la Q-fonction.



b. Politique ϵ-gloutonne.



c. Optimisation directe de la Q-fonction.

FIG. 2 – Chaîne contrôlée.

4.2.2 Politique ϵ-gloutonne

Le problème d'intérêt suivant est l'apprentissage de la Q-function optimale dans un contexte d'itération optimiste de la politique. La politique apprise est ϵ-gloutonne, avec $\epsilon = 0.1$. Le taux

d'apprentissage est le même que précédemment, l'*a priori* est fixé à I et la variance du bruit d'observation (resp. des résidus) à 10^{-3}. Le même bruit d'évolution que dans la section 4.1 est utilisé, avec $\eta = 10^{-3}$. Comme la politique ϵ-gloutonne est aléatoire, nous testons également l'algorithme wXKTD, la différence étant représentée équation (26). Le facteur de corrélation $-\gamma\sigma_i^2$ est ainsi pondéré par $1-\epsilon$ ou ϵ, selon que l'action gloutonne est choisie ou non. L'apprentissage est fait sur 700 épisodes (chacun étant initialisé avec une paire état-action aléatoire et étant limité à 15 interactions), et les résultats présentés sur la figure 2.b sont moyennés sur 100 essais (notons à nouveau l'échelle logarithmique).

KTD et XKTD convergent tous deux plus rapidement dans un premier temps. Ensuite KTD est biaisé, et XKTD converge plus lentement. LSTD converge plus rapidement, mais il est moins stable. Après approximativement 150 épisodes, l'erreur associée augmente. Cela est probablement causé par la non-stationnarité de la politique apprise. Les meilleurs résultats sont obtenus avec l'algorithme wXKTD. Pondérer le facteur de corrélation a donc du sens. Notons qu'il y a toujours du biais, du fait que nous ne faisons pas tendre le facteur ϵ vers 0.

4.2.3 Optimisation directe de la Q-fonction.

Dans cette section, nous proposons d'illustrer le bais causé par l'aspect *off-policy* de l'optimisation directe de la Q-fonction, comme expliqué dans la section 3.3. Comparaison est faite avec le Q-learning. Les paramètres sont les mêmes que dans la section 4.2.2, sans bruit d'évolution. La politique comportementale est totalement aléatoire (tirage uniforme des actions). La mise à jour est faite en accord avec l'équation d'optimalité de Bellman (3). Les résultats sont présentés sur la figure 2.c, l'apprentissage étant fait sur 100 épisodes, et les résultats étant moyennés sur 100 essais.

Quatre algorithmes sont comparés, TD (Q-learning) et (w)(X)KTD. Pour wXKTD, le bruit est coupé chaque fois qu'une action comportementale n'est pas l'action gloutonne. LSTD n'est pas considéré, dans la mesure où ce n'est pas un algorithme du type itération de la valeur. Comme attendu, l'ensemble des algorithmes de la famille KTD converge plus vite, cependant ils sont tous biaisés. KTD est biaisé à cause des transitions stochastiques, XKTD est biaisé en raison de l'aspect *off-policy* du problème considéré, et comme prévu, wXKTD combine les deux sources de biais. Cela n'aide pas à améliorer les performances, contrairement à son application à l'apprentissage d'une politique ϵ-gloutonne de la section 4.2.2.

4.3 *Mountain-car*

La dernière expérience que nous proposons est la tâche du *mountain-car* qui consiste à conduire une voiture sous-puissante hors d'une cuvette. Elle est totalement décrite par Sutton & Barto (1998, chapitre 8.4). L'objectif est ici d'illustrer le comportement de (w)(X)KTD sur un problème de contrôle plus complexe. De façon à le rendre stochastique l'environnement est légèrement modifié : l'action choisie est appliquée avec une probabilité $p = 0.8$, et une des deux autres au hasard avec une probabilité $\frac{1-p}{2}$ chacune. Le facteur d'actualisation est fixé à 0.95. L'état est normalisé, et la paramétrisation est composée d'un terme constant et d'un ensemble de 9 noyaux gaussiens équi-répartis d'écart type 0.5, cela pour chacune des trois actions. Il y a donc un ensemble de 30 fonctions de base.

Cette expérimentation compare TD (SARSA avec approximation de la fonction de valeur), LSTD et (w)(X)KTD dans le cadre d'un schéma d'itération optimiste de la politique. La politique suivie est ϵ-gloutonne, avec $\epsilon = 0.1$. Pour TD, le taux d'apprentissage est fixé à $\alpha = 0.1$. Pour les autres algorithmes, l'*a priori* est fixé à $P_{0|0} = 10I$. Pour (w)(X)KTD la variance du bruit d'observation (resp. la variance des résidus) est choisie égale à 1. Chaque épisode commence dans un état initial uniformément échantillonné. Un maximum de 1500 interactions par épisode est autorisé. Pour chaque essai, l'apprentissage se fait sur 200 épisodes, et la figure 3 montre la longueur de chaque épisode moyennée sur 300 essais.

Toutes les variantes de KTD ont de meilleurs résultats que TD et LSTD sur ce problème stochastique. Comme auparavant, wXKTD se comporte mieux que XKTD. Cependant, de façon assez surprenante, les meilleurs résultats sont obtenus avec KTD, malgré le biais causé par les transitions stochastiques. On peut donc légitimement se demander dans quelle mesure le biais pose problème pour la convergence vers la politique optimale, d'un point de vue de contrôle. De façon générale en

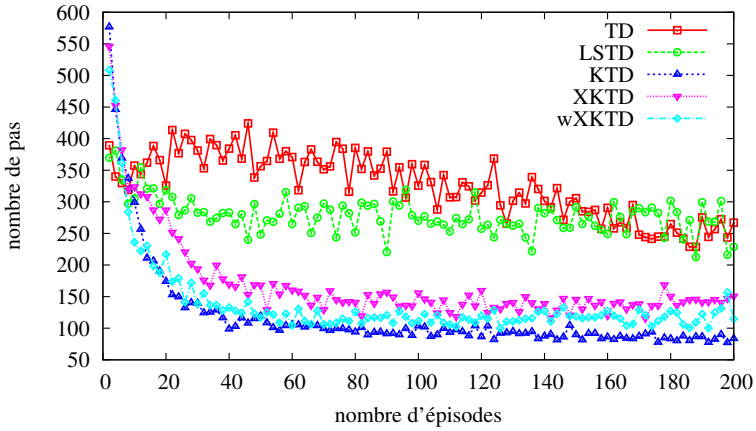


FIG. 3 – *Mountain car.*

apprentissage par renforcement, un bon contrôle ne nécessite pas forcément une estimation précise de la fonction de valeur, et inversement une bonne estimation de la fonction de valeur ne garantie pas nécessairement un bon contrôle.

5 Conclusion

Une extension du cadre de travail des différences temporelles de Kalman, basée sur un modèle de bruit d'observation coloré, au cas des processus décisionnels de Markov stochastiques a été présentée, ainsi qu'une variante pondérée. Elles ont été expérimentées sur un ensemble de tests de référence. Nous avons montré empiriquement que dans le cas de l'évaluation pure, XKTD supprime le biais inhérent à KTD, causé par la stochasticité des transitions. D'un point de vue de contrôle (politique ϵ-gloutonne), wXKTD réduite plus le biais que XKTD. Si l'apprentissage est *off-policy* (et les transitions stochastiques), toutes les variantes de KTD échouent à supprimer le bais, pour les raisons énoncées auparavant (biais causé par les transitions stochastiques, par l'effet mémoriel du bruit coloré, et combinaison des deux). D'un point de vue performance du contrôle, l'avantage des différentes variantes de KTD est moins clair, même si elles se comportent toutes plutôt bien.

Ainsi le cadre de travail de KTD tient la comparaison face aux algorithmes de l'état de l'art et présente certains aspects intéressants, cependant il y a encore quelques points qui méritent d'être développés. Premièrement, l'utilisation du bruit coloré dans un contexte de contrôle, voir *off-policy*, nécessite plus de travail. Il existe une vaste littérature sur le filtrage adaptatif, et nous pensons que KTD peut en profiter. Il pourrait également être intéressant d'utiliser KTD comme critique dans une architecture acteur-critique. Par exemple, des algorithmes acteur-critique incrémentaux basés sur la notion de gradient naturel sont introduits par Bhatnagar *et al.* (2008). TD y est préféré comme critique à LSTD, principalement à cause de l'incapacité de ce dernier à prendre en compte la non-stationnarité. Dans ce cas, KTD pourrait être une alternative intéressante comme critique du second ordre. Nous avons également comme projet d'étendre KTD au cas de l'observabilité partielle, qui devrait pouvoir être assez naturellement implanté dans ce cadre de travail, étant donnée sa nature bayésienne.

Remerciements

Olivier Pietquin souhaite remercier la région Lorraine ainsi que la communauté européenne (projet CLASSiC, FP7/2007-2013, subvention 216594) pour leur support financier.

Références

ANTOS A., SZEPESVÁRI C. & MUNOS R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*,

71(1), 89–129.

Baird L. C. (1995). Residual Algorithms : Reinforcement Learning with Function Approximation. In *Proceedings of the International Conference on Machine Learning (ICML 95)*, p. 30–37.

Bertsekas D. P. & Tsitsiklis J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bhatnagar S., Sutton R. S., Ghavamzadeh M. & Lee M. (2008). Incremental Natural Actor-Critic Algorithms. In *Proceedings of NIPS 21*, Vancouver, Canada.

Boyan J. A. (1999). Technical Update : Least-Squares Temporal Difference Learning. *Machine Learning*, **49**(2-3), 233–246.

Bradtke S. J. & Barto A. G. (1996). Linear Least-Squares Algorithms for Temporal Difference Learning. *Machine Learning*, **22**(1-3), 33–57.

Engel Y. (2005). *Algorithms and Representations for Reinforcement Learning*. PhD thesis, Hebrew University.

Engel Y., Mannor S. & Meir R. (2005). Reinforcement Learning with Gaussian Processes. In *Proceedings of International Conference on Machine Learning (ICML-05)*.

Geist M., Pietquin O. & Fricout G. (2009a). Différences Temporelles de Kalman. In *Journées Françaises Planification Décision Apprentissage (JFPDA 2009)*, Paris, France.

Geist M., Pietquin O. & Fricout G. (2009b). Kalman Temporal Differences : the deterministic case. In *Proceedings of the IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, Nashville, TN, USA.

Julier S. J. & Uhlmann J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, **92**(3), 401–422.

Kalman R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, **82**(Series D), 35–45.

Precup D., Sutton R. S. & Singh S. P. (2000). Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML00)*, p. 759–766, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

Simon D. (2006). *Optimal State Estimation : Kalman, H Infinity, and Nonlinear Approaches*. Wiley & Sons.

Sutton R. S. & Barto A. G. (1998). *Reinforcement Learning : An Introduction*. The MIT Press, 3rd edition.

van der Merwe R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA.

Watkins C. J. (1989). *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, England.